TP à rendre 1 – Code de César

Échéance	le lundi 18 octobre 2010 à 14h
Délivrables	un rapport contenant la réponse aux questions et le code du programme en JAVA à rendre durant la séance d'exercices du lundi 18 octobre 2010. **et** le programme JAVA à poster dans la section travaux sur iCampus

1 Introduction

L'armée a besoin de vos services !

Dans le cadre d'un projet classé top secret, le code ADN du parfait soldat a été découvert dans un des laboratoires de recherche de l'armée. Ce dernier doit être envoyé du laboratoire à l'état major. Pour que personne ne puisse voler le code ADN, celui-ci va être chiffré afin de pouvoir être transmis. En effet, seul le laboratoire de recherche et l'état major doivent avoir connaissance de ce contenu confidentiel. Dans le cadre de ce travail, on va utiliser le *Code de César*[1].

L'ADN est une longue molécule présente dans chaque cellule vivante. Ce dernier joue un rôle notamment dans la synthèse des protéines. L'ADN est composé de deux brins se faisant face et formant une double hélice[2]. Chaque brin est constitué d'une séquence de bases azotées. Il existe quatre bases azotées différentes dans une molécule d'ADN : l'adénine (`A`), la thymine (`T`), la cytosine (`C`) et la guanine (`G`). Les deux brins sont dits complémentaires. C'est-à-dire que si on connait la séquence d'un des brins, on connait automatiquement la séquence de l'autre brin. Une adénine est toujours en regard d'une thymine (et inversément) et une cytosine est toujours en regard d'une guanine (en inversément). Des séquences d'ADN déterminent une protéine qui déterminera le fonctionnement des cellules.

Voici un exemple d'une séquence d'ADN : `"ATTGCCGTATGTATTGCGCT"`.

Le but du travail est de transformer toute séquence d'ADN en une séquence sans signification pour quelqu'un qui déroberait la séquence transformée. Seul le destinataire doit être capable de déchiffrer la séquence transformée pour reconstituer la séquence d'ADN originale.

Il existe diverses techniques permettant de chiffrer des informations, certaines étant meilleures que d'autres. Dans le cadre de ce travail, on ne va pas trop se soucier d'avoir un bon chiffrement et on va travailler avec le Code de César. Il faut deux éléments pour mettre en œuvre cette méthode de chiffrement : un *alphabet* et une *clé*. Ces deux informations doivent être connues par l'expéditeur et le destinataire afin de pouvoir respectivement chiffrer et envoyer le message et le déchiffrer et le lire.

Le principe du Code de César consiste à décaler vers la droite toutes les lettres du message original d'un nombre de position correspondant à la clé. Par exemple, soit l'alphabet $\{A, T, C, G\}$ et soit la clé valant 3. Chaque lettre du message original sera donc décalée de trois positions dans l'alphabet. Ainsi, on peut construire la correspondance suivante $A \rightarrow G, T \rightarrow A, C \rightarrow T$ et $G \rightarrow C$. Prenons par exemple le message `"AAGTC"`, une fois chiffré on obtiendra la séquence `"GGCAT"`. Pour déchiffrer ce message, il suffit de faire l'inverse.

1. http://fr.wikipedia.org/wiki/Chiffre_de_César
2. http://fr.wikipedia.org/wiki/Acide_désoxyribonucléique

2 Travail

On vous demande d'écrire un programme permettant de chiffrer et déchiffrer n'importe quelle *séquence d'ADN* pour n'importe quel clé de chiffrement/déchiffrement. Pour cela, vous n'allez pouvoir utiliser que des objets `chiffre`, `nombreNaturel`, `nombreEntier` et `suiteDeChiffres`. On va travailler avec des suites de chiffres, et utiliser les chiffres `A`, `T`, `C` et `G` pour représenter les chiffres de notre alphabet. Faites par ailleurs bien attention aux préconditions de vos différentes méthodes.

La clé de chiffrement peut avoir n'importe quel valeur entière (positive ou négative) comprise entre -9 et 9. La valeur de la clé indique le décalage des lettres dans l'alphabet. Une valeur positive signifie un décalage vers la droite alors qu'une valeur négative indique un décalage vers la gauche.

Voici les spécifications des deux méthodes que vous devez fournir. Vous êtes bien entendu encouragés à décomposer le problème en sous-problèmes et à refléter cela dans votre programme Java.

```
/**
 * Chiffrer le nombre "msg" en utilisant le chiffrement
 * par code de César avec la clé "cle"
 *
 * PRECONDITION:  - "msg" et "cle" sont initialisés
 * POSTCONDITION: - "msg" n'a pas été modifié
 *                - "cle" n'a pas été modifié
 *                - la méthode renvoie le résultat du chiffrement
 *                  par le Code de César du nombre "msg" avec la clé "cle"
 */
static suiteDeChiffres chiffrer (suiteDeChiffres msg, nombreEntier cle);

/**
 * Déchiffrer le nombre "msg" en utilisant le déchiffrement
 * par code de César avec la clé "cle"
 *
 * PRECONDITION:  - "msg" et "cle" sont initialisés
 * POSTCONDITION: - la méthode renvoie le résultat du déchiffrement
 *                  par code de César du nombre "msg" avec la clé "cle"
 *                - "msg" n'a pas été modifié
 *                - "cle" n'a pas été modifié
*/
static suiteDeChiffres dechiffrer (suiteDeChiffres msg, nombreEntier cle);
```

Votre programme doit également comporter une méthode `main` qui va demander à l'utilisateur ce qu'il désire faire (chiffrer ou déchiffrer), lui demander toutes les informations nécessaires, effectuer l'opération demandée et afficher le résultat de cette dernière. Voici un exemple d'exécution (en rouge les informations entrées par l'utilisateur) :

```
Que voulez-vous faire ?
(1) Chiffrer un message
(2) Déchiffrer un message
(3) Quitter le programme

Votre choix :
1

Quel est votre message ?
AAGTC
Quelle est la clé ?
3
```

```
Le message chiffré est : GGCAT

Que voulez-vous faire ?
(1) Chiffrer un message
(2) Déchiffrer un message
(3) Quitter le programme

Votre choix :
3

Au revoir et merci !
```

Le programme sera documenté, syntaxiquement correct (il compile !) et fonctionnera au moins sur les deux exemples fournis dans la section suivante. Le rapport devra contenir le code source de votre programme en annexe, une discussion de votre code et de la décomposition en sous-problèmes que vous aurez choisie et votre avis sur le code de César.

3 Informations utiles

Pour afficher un objet `suiteDeChiffres` à l'écran, il vous faut écrire la ligne suivante (en supposant que la variable contenant la suite de chiffres est `s`) :

```
ecran.afficher ("" + s);
```

Si vous souhaitez afficher du texte avant la suite de chiffres, insérez-le entre les guillemets, par exemple :

```
ecran.afficher ("Voici ma belle suite de chiffres : " + s);
```

Enfin, bien qu'on ne vous demande de ne supporter que les clés allant de -9 à 9, vous pourriez vouloir généraliser votre solution. Si vous tentez cette généralisation, indiquez le clairement dans votre rapport et expliquez comment vous avez procédé.

4 Rapport

Votre rapport doit contenir une description de la décomposition en sous-problèmes, ainsi que les spécifications de chacun des sous-problèmes. Vous ajouterez également un mode d'emploi expliquant comment compiler et exécuter votre programme.

Pour l'un de vos sous-problème, vous devez montrer qu'il est réutilisable, en donnant un exemple de contexte, différent de ce projet, dans lequel vous pourriez le réutiliser.

Enfin, en une dizaine de lignes, donnez **votre avis**[3] sur le code de César.

3. Il ne s'agit donc pas de faire un copier-coller de Wikipédia.

5 Répartition des points

- Correction syntaxique (le code compile) **(2 points)**

- Lisibilité du code **(4 points)**
  - Mise en forme (indentation, espacements, ...)
  - Nom des méthodes et variables
  - Décomposition en sous-problèmes

- Documentation du code **(5 points)**
  - Commentaires sur le programme (nom, description)
  - Assertions intermédiaires
  - Spécifications des sous-problèmes

- Choix des sous-problèmes **(2 points)**
  - Réutilisabilité (à justifier dans le rapport)

- Fonctionnement sur les exemples (**3 points**)
  1. Chiffrement de la séquence `"AAATCGAG"` avec la clé -2 doit produire `"CCCGATCT"`
  2. Chiffrement de la séquence `"GATTACA"` avec la clé $+9$ doit produire `"ATCCTGT"`
  3. Autres exemples non fournis (dont un déchiffrement)

- Rapport (une dizaine de pages au maximum + le code en annexe) (**4 points**)
  - Orthographe
  - Avis sur le code de César (un paragraphe de 10 lignes maximum)
  - Description générale du code et de la décomposition en sous-problèmes

Bon travail !